process repeats. If, however, another process is not to be executed, as determined in decision block **1110**, processing ends in block **1112**.

Project Maintenance Process

Referring to FIG. **12**, a project maintenance process is illustrated and begins in block **1202**. Thereafter, the user can run a SQL Delete process in block **1204**, a SQL Update process in block **1206**, an Index process in block **1208**, an Archive Project process in block **1210**, and an Un-Archive process in block **1212**. After the completion of the selected process, the user can execute another process as determined in decision block **1214**. If another process is to be executed, processing loops back and the above described process repeats. If, however, another process is not to be executed, as determined in decision block **1214**, processing ends in block **1216**.

Communication Between External Applications and Processing Agents

Referring now to FIG. **13**, the communication process between an external application **1302**, which can be web-based, and a processing agent **1304** is illustrated. The external application **1302** establishes a connection **1308** with the ServerLookupAgent **1306** using a known port number assigned to the ServerLookup Agent **1306**. The external application **1302** then sends a port number request **1310** for a client specific InteractiveServerAgent **1314** to the ServerLookupAgent **1306**. The ServerLookupAgent **1306** sends a response **1312** that contains the port number for the requested client specific InteractiveServerAgent **1314** to the external application **1302**. Thereafter, the external application **1302** sends all future requests **1316** to the InteractiveServerAgent **1314**. When the InteractiveServerAgent **1314** receives a request **1316**, it sends a "ticket" request to the agent manager **1320** so that the InteractiveServerAgent **1314** can communicate with the processing agent **1304**. The agent manager **1302** gets the processing agent **1304** ready to "speak" with the InteractiveServerAgent **1314** and then sends the ticket **1322** to the InteractiveServerAgent **1314**. The InteractiveServerAgent **1314** then redeems the ticket and begins speaking with the processing agent **1304** via requests **1324**. The processing agent **1304** sends the responses **1326** back through the InteractiveServerAgent **1314**, which then forwards those responses **1328** to the external application **1302**.

Now referring also to FIG. **14**, a block diagram of the InteractiveServerAgent **1314** process is illustrated and begins in block **1402** when the InteractiveServerAgent **1314** is launched. The InteractiveServerAgent **1314** requests an open socket connection to the ServerLookupAgent **1306** in block **1404**, which is supplied by the operating system. The InteractiveServerAgent **1314** then registers the port number corresponding to the open socket connection and client name with the ServerLookupAgent **1306** in block **1406**. Thereafter, the InteractiveServerAgent **1314** processes all incoming requests **1316** and responses **1326** to and from the external application **1302** and the appropriate processing agent **1304**. If an incoming request **1316** has not been received, as determined in block **1408**, and an incoming response **1326** has not been received, as determined in block **1410**, and the InteractiveServerAgent **1314** is not to be terminated, as determined in block **1412**, the process loops back to decision block **1408** and continues to look for incoming requests **1316**, incoming responses **1326** or an instance termination signal.

Whenever an incoming request **1316** is received, as determined in decision block **1408**, the request **1316** is parsed using the agent manager **1320** in block **1414** and then passed to the appropriate processing agent **1304** in block **1416**. The process then continues to look for incoming requests **1316**, incoming responses **1326** or an instance termination signal as previously described. Similarly, whenever an incoming response **1326** is received, as determined in decision block **1410**, the response **1326** is passed to the external application **1302** in block **1418**. The process then continues to look for incoming requests **1316**, incoming responses **1326** or an instance termination signal as previously described. If an instance termination signal is received, as determined in decision block **1412**, the InteractiveServerAgent **1312** unregisters the port number and client name with the ServerLookupAgent **1306** in block **1420**, closes the socket connection to the ServerLookupAgent **1306** in block **1422** and terminates the instance of the InteractiveServerAgent **1312** in block **1424**. Note that the InteractiveServerAgent **1312** runs continuously until the instance or thread of the InteractiveServerAgent **1312** is terminated. Typically, such a termination only occurs when the InteractiveServerAgent **1312** is taken down by hand.

As previously described, the ServerLookupAgent **1306** and InteractiveServerAgent **1312** allow multiple external applications **1302** to simultaneously connect with any client within the system. Any client in the system that desires to support this interactive interface will run an instance of the InteractiveServerAgent **1312**. Each client will also run Agent instances for the various interactive functions it wishes to provide. Note that the client does not have to provide all possible functions or services. Accordingly, the set of processing Agents that are available are defined in the client's configuration file. Moreover, each agent participating in the interactive interface is designed to handle many requests from any external application **1302**.

Although preferred embodiments of the invention have been described in detail, it will be understood by those skilled in the art that various modifications can be made therein without departing from the spirit and scope of the invention as set forth in the appended claims.

What is claimed is:

1. An apparatus for managing data comprising:

a first database for storing system data;

a second database for storing client data;

at least one database server to control the first database and the second database;

an external application;

a first set of objects linked to the first database through the database server;

a second set of objects linked to the second database through the database server;

an interactive manager linked to the first set of objects and the external application;

a set of minions linked to the second set of objects; and

an agent manager linked to the interactive manager, the first set of objects and the set of minions.

2. The apparatus as recited in claim **1**, wherein the client data is marketing data.

3. The apparatus as recited in claim **1**, wherein the interactive manager is a first agent process and the agent manager is a second agent process.

4. The apparatus as recited in claim **3**, wherein each agent process further comprises:

an agent;